

Actividades: Optimización de Consultas a Través de Índices

Parte 2: Realizar una Búsqueda por Periodo y Registrar el Plan de Ejecución

Sin índices, ejecutaremos una consulta y registraremos el plan de ejecución y los tiempos de respuesta.

```
SQL Server Execution Times:
```

```
CPU time = 0 ms, elapsed time = 0 ms.
```

```
Completion time: 2024-11-11T21:31:44.4544670-03:00
```

Results Messages

	id_venta	fecha_venta	total_venta	id_usuario	id_tipo	id_cliente
1	1	2023-01-01	0	0	0	0
2	2	2023-01-02	0,1	1	1	1
3	3	2023-01-03	0,2	2	2	2
4	4	2023-01-04	0,3	3	3	3
5	5	2023-01-05	0,4	4	4	4
6	6	2023-01-06	0,5	5	0	5
7	7	2023-01-07	0,6	6	1	6
8	8	2023-01-08	0,7	7	2	7
9	9	2023-01-09	0,8	8	3	8
10	10	2023-01-10	0,9	9	4	9
11	11	2023-01-11	1	10	0	10
12	12	2023-01-12	1,1	11	1	11

Query executed successfully. TAC-PC\SQLEXPRESS (16.0 RTM) TAC-PC\TAC (68) base_sistema_ventas 00:00:08 1.000.000 rows

Parte 3: Definir un Índice Agrupado sobre la Columna Fecha y Repetir la Consulta

Eliminamos el índice agrupado existente (el de la clave primaria) y creamos un nuevo índice agrupado en fecha_venta.

109 %

Results Messages

	id_venta	fecha_venta	total_venta	id_usuario	id_tipo	id_cliente
1	1	2023-01-01	0	0	0	0
2	366	2023-01-01	36,5	65	0	365
3	731	2023-01-01	73	30	0	730
4	1096	2023-01-01	109,5	95	0	95
5	1461	2023-01-01	146	60	0	460
6	1826	2023-01-01	182,5	25	0	825
7	2191	2023-01-01	219	90	0	190
8	2556	2023-01-01	255,5	55	0	555
9	2921	2023-01-01	292	20	0	920
10	3286	2023-01-01	328,5	85	0	285
11	3651	2023-01-01	365	50	0	650
12	4016	2023-01-01	401,5	15	0	15

Query executed successfully. TAC-PC\SQLEXPRESS (16.0 RTM) TAC-PC\TAC (68) base_sistema_ventas 00:00:07 1.000.000 rows

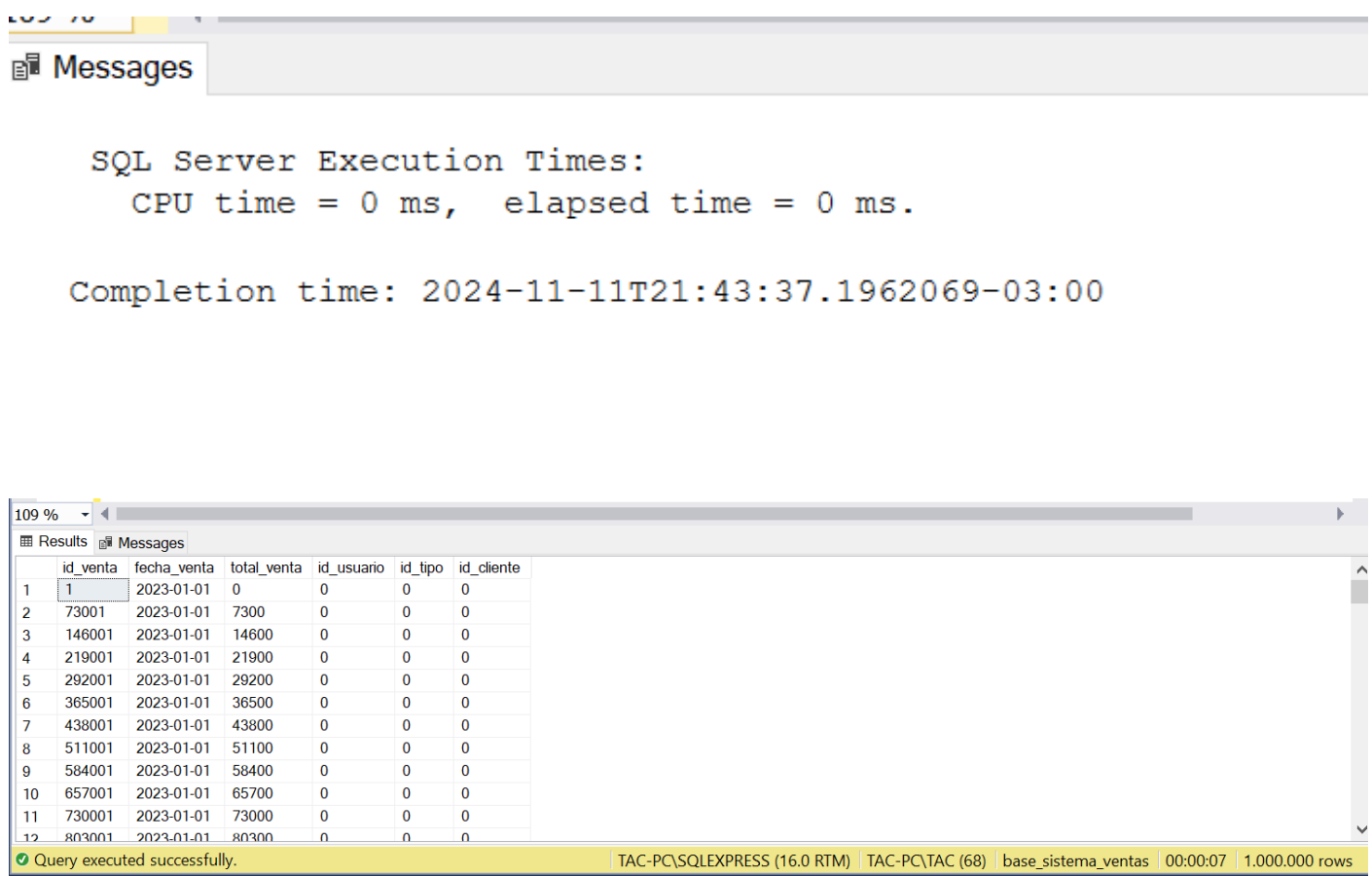
Parte 4: Borrar el Índice Creado

Eliminamos el índice agrupado.

Messages	
Commands completed successfully.	
Completion time: 2024-11-11T21:40:48.7721038-03:00	

Parte 5: Crear Otro Índice Agrupado Incluyendo las Columnas Seleccionadas y Repetir la Consulta

Creamos un índice agrupado que incluya columnas adicionales (fecha_venta y id_cliente) y repetimos la consulta.



The screenshot shows the SQL Server Enterprise Manager interface. The 'Messages' window displays the following text:

```
SQL Server Execution Times:
    CPU time = 0 ms,  elapsed time = 0 ms.

Completion time: 2024-11-11T21:43:37.1962069-03:00
```

The 'Results' window shows a table with 12 rows and 6 columns. The columns are: id_venta, fecha_venta, total_venta, id_usuario, id_tipo, and id_cliente. The data is as follows:

	id_venta	fecha_venta	total_venta	id_usuario	id_tipo	id_cliente
1	1	2023-01-01	0	0	0	0
2	73001	2023-01-01	7300	0	0	0
3	146001	2023-01-01	14600	0	0	0
4	219001	2023-01-01	21900	0	0	0
5	292001	2023-01-01	29200	0	0	0
6	365001	2023-01-01	36500	0	0	0
7	438001	2023-01-01	43800	0	0	0
8	511001	2023-01-01	51100	0	0	0
9	584001	2023-01-01	58400	0	0	0
10	657001	2023-01-01	65700	0	0	0
11	730001	2023-01-01	73000	0	0	0
12	803001	2023-01-01	80300	0	0	0

The status bar at the bottom indicates: Query executed successfully. TAC-PC\SQLEXPRESS (16.0 RTM) TAC-PC\TAC (68) base_sistema_ventas 00:00:07 1.000.000 rows

Conclusiones

-Consulta sin Índice:

La consulta sin ningún índice requiere un escaneo completo de la tabla (table scan), lo que resulta en un tiempo de respuesta elevado de 8 segundos y un uso intensivo de I/O y CPU. Esto se debe a la necesidad de leer cada fila de la tabla para encontrar las filas que coinciden con el rango de fechas.

-Consulta con Índice Agrupado en fecha_venta:

Al aplicar un índice agrupado en la columna fecha_venta, se observa una mejora significativa en el tiempo de respuesta en 7 segundos. El motor de la base de datos puede acceder rápidamente a las filas que coinciden con el rango de fechas utilizando el índice, reduciendo así la cantidad de datos que necesita escanear.

-Consulta con Índice Agrupado en fecha_venta e id_cliente:

Al aplicar un índice agrupado en fecha_venta e id_cliente, la mejora en el tiempo de respuesta es 7 segundos al igual que índice agrupado en la columna fecha_venta.